Honey Cloud: A Honeypot Network Approach for Enhanced Security to the Cloud

Sammeta Chandana[1], Katakam Sai Kumar[2], Pallapu Spoorthi[3], Nageswara Rao. M[4]
[1,2,3] UG Scholor , Dept.of AI & ML , St Martin's Engineering College , Secunderabad , Telangana , India , 500100
[4]Assistant Professor, Dept.of AI & ML, St Martin's Engineering College, Secunderabad, Telangana, India, 500100
sammeta.chandana@gmail.com

Abstract:

The rapid increase in the number of users across digital platforms has led to a surge in challenges related to hardware failure, web hosting, and the efficient allocation of data storage and memory. These issues, if left unaddressed, can result in the loss of valuable data. To overcome these limitations and provide reliable, fast, and cost-effective services, many organizations have turned to cloud computing. Cloud computing offers scalable infrastructure that can handle growing user demands and data requirements. However, with the exponential growth of cloud-based systems, the risk of malicious cyberattacks and data breaches has also escalated, leading to concerns regarding the security of sensitive data hosted on the cloud. In response to these security challenges, the use of honeypots has emerged as an effective strategy for diverting malicious traffic away from critical systems. A honeypot is a decoy system designed to appear vulnerable, luring attackers into engaging with it instead of the actual production system. By analyzing the activities of malicious users, honeypots provide valuable insights into attack patterns and help enhance the security of real systems. This project focuses on the implementation of a honeypot-based security solution within a file-sharing application deployed on a cloud server. The goal is to explore how this technology can be integrated into cloud-hosted applications to safeguard against potential cyber threats while maintaining system integrity. The project will also address the legal and ethical implications of deploying honeypots on third-party cloud vendor servers, considering privacy concerns, data protection laws, and the possible risks associated with such deployments. By examining the potential of honeypots in a cloud-based environment, this study seeks to contribute to the development of more secure, efficient, and cost-effective cloud services, while mitigating the risks posed by malicious actors. Additionally, the project will explore the design and operational challenges of incorporating honeypots into cloud-based file-sharing systems, including performance impacts, legal considerations, and the overall feasibility of such an approach.

Keywords: Users, digital platforms, hardware failure, web hosting, data storage, memory allocation, data loss, cloud computing, scalable infrastructure, user demands.

1.INTRODUCTION

Cloud computing is a technique to store, share and access data anytime and anywhere with a device that is connected to a network, preferably the internet. Cloud computing consists of an expandable storage space with no physical storage space which is accessible from anywhere in the world using any device, by connecting it to the internet. It contains large number of computing devices connected through a real-time communication (the internet) and has a common data storage area. The term "the cloud" is used as a metaphor for the Internet, based on the fact that a cloud like shape was used to indicate network telephone schematics, and later the Internet as an abstraction of underlying infrastructure it represents. Honeypots are viewed as a successful technique to track programmer conduct and uplift the viability of security instruments. Honeypots are specifically designed to not only purposely engage and deceive hackers but also identify malicious activities performed over the Internet and can be counted as an effective method to track hacker behaviour. Honeypots can be defined as systems or assets which are used to not only trap, monitor but to also identify erroneous requests present within a network. They vary in the interaction provided to the attackers, from low interaction to medium and high, each type has its advantages and disadvantages. Their aim is to analyze, understand, watch and track attacker's behaviour in order to create systems that are not only secure but can also handle such traffic. It is a closely monitored computing resource that we want to be probed, attacked, or compromised. "More precisely, it is an information system resource whose value lies in unauthorized or illicit use of that resource.".

Cloud-based Honeypots give the capacity to explore and examine assaults that hit ordinary customers. Having them permits a specialist to interpret the IP locations and malware being utilized into security content that can ensure a normal cloud environment. Once those IP addresses have been distinguished, they will then lead a ping scope and defencelessness output to discover a shortcoming in the system outline or vulnerabilities in programming that can be misused. It's conspicuous yet genuine; awful folks pursue the weakest focuses the most often. There are upsides of utilizing a cloud construct Honeypot in light of a cloud framework is like customary Honeypots in that it ought to have the capacity to decide whether a cloud framework has been traded off or endeavours were made to do so. At last, they can essentially sit and log all movement coming into the cloud site; and in light of the fact that it's utilized for this particular reason practically any action ought to be dealt with as instantly suspicious. Honeypots can serve to make dangers more obvious and go about as an early alert framework, which gives a cloud organization a more proactive way to deal with security instead of responsive. Any association with either outside resources/areas or cloud administrations ought to deploy cloud-based Honeypots.

2. LITERATURE SURVEY

[1] Honeypot forensics for system and network SIEM design

Jeremy Briffaut, Clemente Patrice, Jean-Franc Lalande, Jonathan Rouzaud-Cornabas,2003

This chapter presents forensic investigations of cyber attackers' activities on a large scale honeypot and shows how these methodologies can be integrated into an SIEM. The chapter describes our high interaction honeypot and analyzes the illegal activities performed by attackers on the basis of the data collected over two years of attacks: logged sessions, intrusion detection system alerts, mandatory access control system alerts. The empirical study of these illegal activities has allowed us to understand the global motivations of the attackers, their technical skills, the geographical location of the attackers and their targets. A generic method is presented that has enabled us to rebuild the illegal activities using correlation techniques operating on system and network events. Monitoring the network and the operations occurring on each system has provided precise and high level characterization of attacks. Finally, the chapter explains how network and system methods for forensics can be integrated into an SIEM in order to more accurately monitor the security of a pool of hosts.

[2] The Nepenthes Platform: An Efficient Approach to Collect Malware

Paul Baecher,Markus Koetter,Thorsten Holz,Felix C. Freiling,2006, there is little empirically backed quantitative and qualitative knowledge about self-replicating malware publicly available. This hampers research in these topics because many counter-strategies against malware, e.g., network- and host-based intrusion detection systems, need hard empirical data to take full effect. We present the nepenthes platform, a framework for large-scale collection of information on self-replicating malware in the wild. The basic principle of nepenthes is to emulate only the vulnerable parts of a service. This leads to an efficient and effective solution that offers many advantages compared to other honeypot-based solutions. Furthermore, nepenthes offers a flexible deployment solution, leading to even better scalability. Using the nepenthes platform we and several other organizations were able to greatly broaden the empirical basis of data available about self-replicating malware and provide thousands of samples of previously unknown malware to vendors of host-based IDS/anti-virus systems. This greatly improves the detection rate of this kind of threat.

[3] HoneyLab: Large-Scale Honeypot Deployment and Resource Sharing

W.Y. Chin**,**Evangelos P. Markatos,Spiros Antonatos,Sotiris Ioannidis, 2009**.**Honeypots are valuable tools for detecting and analyzing malicious activity on the Internet. Successful and time-critical detection of such activity often depends on large-scale deployment. However, commercial organizations usually do not share honeypot data, and large, open honeypot initiatives only provide read-only alert feeds. As a result, while large and resourceful organizations can afford the high cost of this technology, smaller security firms and security researchers are fundamentally constrained. We propose and build a shared infrastructure for deploying and monitoring honeypots, called HoneyLab, that is similar in spirit to PlanetLab. With an overlay and distributed structure of address space and computing resources, HoneyLab increases coverage and accelerates innovation among security researchers as well as security industry experts relying on honeypot-based attack detection technology. Unlike current honeypot infrastructures, HoneyLab allows security firms and security researchers to deploy their own honeypot services, instrumentation code, and detection algorithms, dispensing the need for setting up a separate honeypot infrastructure whenever a new attack detection method needs to be deployed or tested.

[4] SYNEMA: Visual Monitoring of Network and System Security Sensors

Aline Bousquet,Clemente Patrice,Jean-François Lalande, 2011**.**This paper presents a new monitoring tool called SYNEMA that helps to visualize different types of alerts from well-known security sensors. The architecture of the proposed tool is distributed and enables centralizing the collected information into a lightweight visualizer. The front-end proposes many display modes in order to give the ability to clearly see malicious activities and to be able to visually monitor information collected at system, network and user level in the hosts. The paper concludes with development perspectives about an auto-configurable plugin for visual correlation of attacks.

[5] A Virtual Honeypot Framework

Niels Provos,2004

A honeypot is a closely monitored network decoy serving several purposes: it can distract adversaries from more valuable machines on a network, provide early warning about new attack and exploitation trends, or allow in-depth examination of adversaries during and after exploitation of a honeypot. Deploying a physical honeypot is often time intensive and expensive as different operating systems require specialized hardware and every honeypot requires its own physical system. This paper presents Honeyd, a framework for virtual honeypots that simulates virtual computer systems at the network level. The simulated computer systems appear to run on unallocated network addresses. To deceive network fingerprinting tools, Honeyd simulates the networking stack of different operating systems and can provide arbitrary routing topologies and services for an arbitrary number of virtual systems. This paper discusses Honeyd's design and shows how the Honey framework helps in many areas of system security, e.g. detecting and disabling worms, distracting adversaries.

[6] The Eucalyptus Open-Source Cloud-Computing System

Cloud computing systems fundamentally provide ac- cess to large pools of data and computational resources through a variety of interfaces similar in spirit to exist- ing grid and HPC resource management and program- ming systems. These types of systems offer a new pro- gramming target for scalable application developers and have gained popularity over the past few years. However, most cloud computing systems in operation today are pro- prietary, rely upon infrastructure that is invisible to the research community, or are not explicitly designed to be instrumented and modified by systems researchers. In this work, we present EUCALYPTUS - an open- source software framework for cloud computing that im- plements what is commonly referred to as Infrastructure as a Service (IaaS); systems that give users the ability to run and control entire virtual machine instances deployed across a variety physical resources. We outline the ba- sic principles of the EUCALYPTUS design, detail impor- tant operational aspects of the system, and discuss archi- tectural trade-offs that we have made in order to allow Eucalyptus to be portable, modular and simple to use on infrastructure commonly found within academic settings. Finally, we provide evidence that EUCALYPTUS enables users familiar with existing Grid and HPC systems to ex- plore new cloud computing functionality while maintain- ing access to existing, familiar application development software and Grid middle-ware.

[7] SGNET: A Worldwide Deployable Framework to Support the Analysis of Malware Threat Models

Corrado Leita,Marc Dacier, 2008.The dependability community has expressed a growing interest in the recent years for the effects of malicious, external, operational faults in computing systems, ie. intrusions. The term intrusion tolerance has been introduced to emphasize the need to go beyond what classical fault tolerant systems were able to offer. Unfortunately, as opposed to well understood accidental faults, the domain is still lacking sound data sets and models to offer rationales in the design of intrusion tolerant solutions. In this paper, we describe a framework similar in its spirit to so called honey-farms but built in a way that makes its large-scale deployment easily feasible. Furthermore, it offers a very rich level of interaction with the attackers without suffering from the drawbacks of expensive high interaction systems. The system is described, a prototype is presented as well as some preliminary results that highlight the feasibility as well as the usefulness of the approach.

[8] Honeypots: Sticking it to hackers

L. Spitzner,2003.Various aspects of honeypots, a security resource whose value lies in being probed, attacked or compromised, are discussed. It is found that production honeypots are used to secure the organization by either preventing, detecting or assisting in the response to an attack. The analysis showed that research honeypots are more complex and more risky than production honeypots.

[9] Sensor in the Dark: Building Untraceable Large-Scale Honeypots Using Virtualization Technologies

Akihiro Shimoda,Tatsuya Mori,Shigeki Goto,2010. A Honeypot is a system that aims to detect and analyze malicious attacks attempted on a network in an interactive manner. Because the primary objective of a honeypot is to detect enemies without being known to them, it is important to hide its existence. However, as several studies have reported, exploiting the unique characteristics of hosts working on a consecutive IP addresses range easily reveals the existence of honeypots. In fact, there exist some anti-honeypot tools that intelligently probe IP address space to locate Internet security sensors

including honeypots. In order to tackle this problem, we propose a system called DarkPots, that consists of a large number of virtualized honeypots using unused and nonconsecutive IP addresses in a production network. DarkPots enables us to deploy a large number of honeypots within an active IP space used for a production network; thus detection is difficult using existing probing techniques. In addition, by virtually classifying the unused IP addresses into several groups, DarkPots enables us to perform several monitoring schemes simultaneously. This function is meaningful because we can adopt more than one monitoring schemes and compare their results in an operating network. We design and implement a prototype of DarkPots and empirically evaluate its effectiveness and feasibility by concurrently performing three independent monitoring schemes in a high-speed campus network. The system successfully emulated 7,680 of virtualized honeypots on a backbone link that carries 500 Mbps - 1 Gbps of traffic without affecting legitimate traffic. Our key findings suggest: (1) active and interactive monitoring schemes provide more in-depth insights of malicious attacks, compared to passive monitoring approach in a quantitative way, and (2) randomly distributed allocation of IP addresses has an advantage over the concentrated allocation in that it can collect more information from malwares.

[10] A Break in the Clouds: Towards a Cloud Definition

Luis Vaquero,Luis Rodero-Merino,Juan Cáceres,Maik Lindner, 2009

This paper discusses the concept of Cloud Computing to achieve a complete definition of what a Cloud is, using the main characteristics typically associated with this paradigm in the literature. More than 20 definitions have been studied allowing for the extraction of a consensus definition as well as a minimum definition containing the essential characteristics. This paper pays much attention to the Grid paradigm, as it is often confused with Cloud technologies.

3. PROPOSED METHODOLOGY

In this project we are developing Honeypot server to detect and prevent attacks. Honeypot is an additional server which sit between user and cloud server and whenever any user send request then Honeypot will intercept that request and authenticate user and his request and if user authenticated then it allow user to access cloud server and if user is not authenticated and send request with fake password then Honeypot will monitor all his activity and serve him fake responses and the attacker with think he successfully attack server and keep sending malicious activities and Honeypot will record all his activity and later admin will block such IP address or instruct server from serving such IP Address.
In propose work designing Honeypot server which accept user request to upload, download and share file. While sharing file users will gave sharing permission and password to genuine users and then shared users can give password to download file. If any malicious user try to download file with fake password then Honeypot server will serve him blank page.
In propose paper if we serve blank page to attacker then he will easily understand that Honeypot serving him empty response then he will stop doing further activity and Honeypot cannot extract more information him and to avoid this problem in extension work instead of serving empty response Honeypot serve fake file which assure attacker that server has successfully hijack and he continue sending malicious activity which help Honeypot extract more information from him.

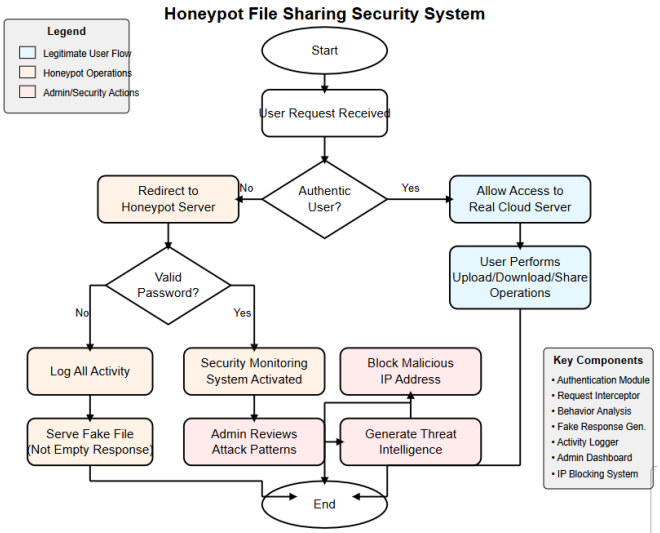

Figure 1: Proposed Model

The proposed methodology typically includes the following key components:

Request Interception: The Honeypot server intercepts all incoming user requests, analyzing them to determine whether the request is legitimate or potentially malicious.

Authentication System: A robust authentication mechanism is implemented where the server checks the provided password. If the user provides a correct password, they are allowed access to the cloud server. If the password is incorrect or suspicious, the Honeypot server will monitor their actions.

Fake Response Generation: If a malicious user attempts unauthorized access, the Honeypot serves fake responses. Instead of serving an empty page, the Honeypot delivers a fake file (or some other type of fake content) to deceive the attacker and encourage further activity.

Activity Monitoring and Logging: The Honeypot logs all interactions, including failed authentication attempts, suspicious file access attempts, and other potentially malicious behavior. This data is used to analyze the attacker's methods.

IP Blocking System: After gathering enough information on the attacker, the Honeypot system can block malicious IP addresses, preventing further access to the server and protecting the cloud infrastructure.

Permission-Based File Sharing: Users can share files with others by setting permissions and providing passwords. The Honeypot ensures that only those with the correct credentials can access the shared files, while malicious users attempting unauthorized access are dealt with by the system.

Admin Control and Alerts: Administrators receive alerts and have the ability to manually intervene in case of suspicious activity. The system allows easy management of IP blocking and other security actions.

Applications:

Cloud Security: The Honeypot can be used to enhance the security of cloud-based services by acting as an intermediary between users and the actual cloud server. It helps identify and prevent attacks before they reach the main server.

Cyber Threat Intelligence: The data collected by the Honeypot can be used to build comprehensive intelligence about cyber threats, attack strategies, and techniques. This helps organizations stay ahead of potential attackers by analyzing attack trends and improving defenses accordingly.

Testing and Research: Researchers can use the Honeypot system for testing new attack techniques, evaluating new defense mechanisms, and studying attacker behaviors. It also aids in testing the effectiveness of various security measures.

Malware Analysis: The system can be used to identify and analyze malware by monitoring its interaction with the Honeypot. This can be useful in studying the tactics of malware authors and improving anti-malware systems.

Data Leak Prevention: By monitoring file-sharing activities and detecting unauthorized attempts to access shared files, the Honeypot helps in preventing potential data leaks and safeguarding sensitive information.

Forensic Analysis: In case of an attack on a real system, the data captured by the Honeypot can be used for forensic investigation, helping administrators understand the attacker's methods and identify any vulnerabilities in the actual server.

Intrusion Detection Systems (IDS): Honeypots can act as part of an Intrusion Detection System, complementing traditional IDS by identifying and tracking unauthorized access attempts and providing real-time alerts for administrators.

Advantages:

Request Interception: The Honeypot server intercepts all incoming user requests, analyzing them to determine whether the request is legitimate or potentially malicious.

Authentication System: A robust authentication mechanism is implemented where the server checks the provided password. If the user provides a correct password, they are allowed access to the cloud server. If the password is incorrect or suspicious, the Honeypot server will monitor their actions.

Fake Response Generation: If a malicious user attempts unauthorized access, the Honeypot serves fake responses. Instead of serving an empty page, the Honeypot delivers a fake file (or some other type of fake content) to deceive the attacker and encourage further activity.

Activity Monitoring and Logging: The Honeypot logs all interactions, including failed authentication attempts, suspicious file access attempts, and other potentially malicious behavior. This data is used to analyze the attacker's methods.

IP Blocking System: After gathering enough information on the attacker, the Honeypot system can block malicious IP addresses, preventing further access to the server and protecting the cloud infrastructure.

Permission-Based File Sharing: Users can share files with others by setting permissions and providing passwords. The Honeypot ensures that only those with the correct credentials can access the shared files, while malicious users attempting unauthorized access are dealt with by the system.

Admin Control and Alerts: Administrators receive alerts and have the ability to manually intervene in case of suspicious activity. The system allows easy management of IP blocking and other security actions.

4. EXPERIMENTAL ANALYSIS

In this project we are developing Honeypot server to detect and prevent attacks. Honeypot is an additional server which sit between user and cloud server and whenever any user send request then Honeypot will intercept that request and authenticate user and his request and if user authenticated then it allow user to access cloud server and if user is not authenticated and send request with fake password then Honeypot will monitor all his activity and serve him fake responses and the attacker with think he successfully attack server and keep sending malicious activities and Honeypot will record all his activity and later admin will block such IP address or instruct server from serving such IP Address. In propose work designing Honeypot server which accept user request to upload, download and share file. While sharing file users will gave sharing permission and password to genuine users and then shared users can give password to download file. If any malicious user try to download file with fake password then Honeypot server will serve him blank page.In propose paper if we serve blank page to attacker then he will easily understand that Honeypot serving him empty response then he will stop doing further activity and Honeypot cannot extract more information him and to avoid this problem in extension work instead of serving empty response Honeypot serve fake file which assure attacker that server has successfully hijack and he continue sending malicious activity which help Honeypot extract more information from him.

Figure 2: Register Operation

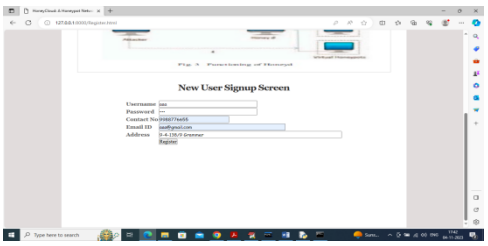

Figure 3: User Signup

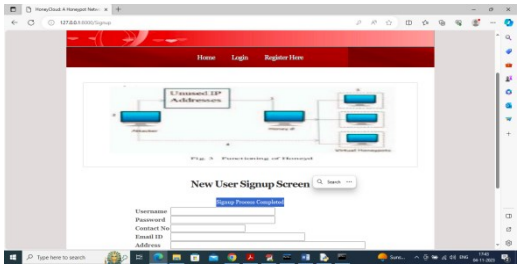

Figure 4: Signup Process Completion

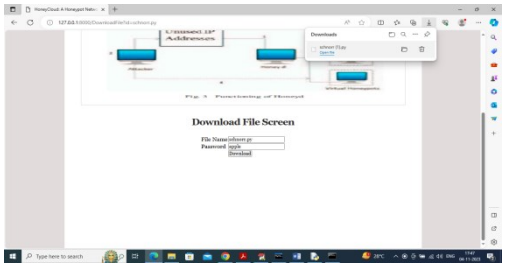

Figure 5: Downloading File

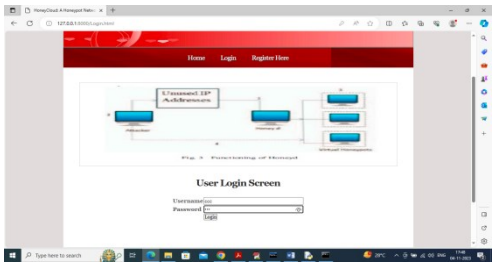

Figure 6 : Files List

Figure 7 : Download Completion

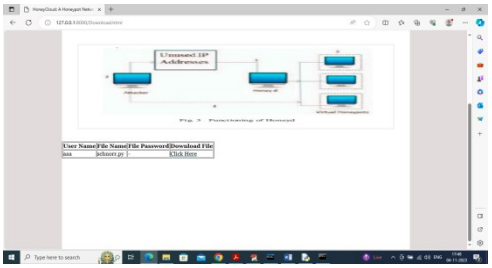

Figure 8 : Login Operation

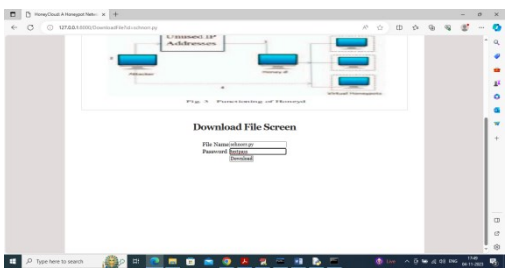

Figure 9 : Downloading File

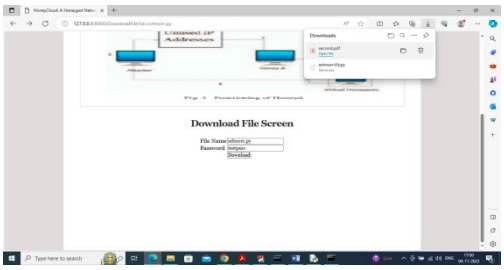

Figure 10 : Result

5. CONCLUSION

Any Organization or firm with either outside resources/areas or cloud administrations ought to deploy cloud-based Honeypots. The IT staff might be required to arrange the Honeypots, yet the genuine outline ought to be driven by the security groups will's identity observing for vindictive movement. Any association managing delicate information in the cloud must prefer Honeypots, and they will likewise require talented system heads to screen the logs and respond to the information. There are some incredible open source apparatuses that have been created to help with the observing and log gathering of Honeypots. It clearly relies on the cloud stage itself. "The perfect Honeypot for Amazon EC2 will contrast from Microsoft's Azure or IBM's cloud". In some ways, the customary Honeypots are not perfect as they tend to reflect the more conventional desktop and server working frameworks. They are, be that as it may, definitely best conveyed where fitting security experts are likewise checking and breaking down at all circumstances. The supplementary utilization of human collaboration gives that additional layer of security and the expert may distinguish a potential or hurtful assault that had never been seen and henceforth observing programming would have no learning." One of the best bits of best practice counsel is to redo from the get go. Honeypot innovation is open source thus the awful folks will be exceptionally acquainted with default settings and will screen for these early signs of a trap. These systems must be setup in an environment which care about their customers and want an extra edge in security in their cloud based platform. Cloud is one of the few technologies that can bring about a major change, hence it is very necessary to make security of cloud more strong. In this paper we present a way to tackle malicious users using Honeypot. Organizations can prefer using Honeypot for detection of rogue elements. One can easily understand the behavior of an attacker by implementing it. Since risks are increasing day by day in Information. Technology extra efforts are required to be put in. Honeypot ensures extra security and detection feature which can be further increased in standard as the technology advances.

REFERENCES

[1] RuWei Huang, Si Yu, Wei Zhuang and XiaoLinGui, “Design of PrivacyPreserving Cloud Storage Framework 2010 Ninth International Conference on Grid and Cloud Computing.

[2] Lizhe Wang, Jie Tao, Kunze M., Castellanos A.C., Kramer D., Karl W., “Scientific Cloud Computing: EarlyDefinition and Experience,” 10th IEEE Int. Conference onHigh Performance Computing and Communications, pp. 825- 830, Dalian, China, Sep. 2008

[3] Cong Wang, Qian Wang, KuiRen and Wenjing Lou, “Ensuring Data Storage Security in Cloud Computing”, InQuality of Service, 2009. 17th International Workshop on, page 19, 2009.

[4] Cong Wang, Qian Wang, Kui Ren and Wenjing Lou “Ensuring Data Storage Security in Cloud Computing.” IEEE 2009.

[5] Balachandra Reddy Kandukuri, Rama Krishna Paturi and Dr. AtanuRakshit, "Cloud security issues" In ServicesComputing, 2009. IEEE International Conference on, page 517520, 2009.

[6] Kashish Goyal, SupriyaKinger" Modified Caesar Cipher for Better Security Enhancement" International Journal ofComputer Applications (0975– 8887) Volume 73– No.3, July 2013.

[7] Yogesh Kumar, Rajiv Munjal and Harsh Sharma,"Comparison of Symmetric and Asymmetric Cryptography withExisting Vulnerabilities and Countermeasures" IJCSMS International Journal of Computer Science and ManagementStudies, Vol. 11, Issue 03, Oct 2011.

[8] Mr. Gurjeevan Singh, Mr. AshwaniSingla and Mr. K S Sandha " Cryptography Algorithm Compaison ForSecurity Enhancement In Wireless Intrusion Detection System" International Journal of Multidisciplinary ResearchVol.1 Issue 4, August 2011.

[9] D. S. Abdul. Elminaam, H. M. Abdul Kader and M. M. Hadhoud ," Performance Evaluation of SymmetricEncryption Algorithms", Communications of the IBIMA Volume 8, 2009.

[10] Gurpreet Singh, SupriyaKinger"Integrating AES, DES, and 3 - DES Encryption Algorithms for Enhanced DataSecurity "International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013.

[11] Uma Somani, "Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security ofCloud in Cloud Computing," 2010 1st International Conference on Parallel, Distributed and Grid Computing (PDGC- 2010).

[12] ChitraRajagopalan. P,TanupriyaChoudhury,Praveen Kumar, A Proposal and Implementation of Algorithm to enhance the security of the cloud", 5th Fifth International Conference on System Modeling & Advancement in Research Trends,IEEE,2016.

[13] BhaskarMandal,Tanupriya Choudhury," A Key Agreement Scheme for Smart Cards Using Biometrics.", IEEE International Conference (Published in IEEE) ICCCA 2016 ,Galgotias University,2016.

[14] Bhaskar Mandal ,Tanupriya Choudhury, "A Secure Biometric Image Encryption Scheme using Chaos and Wavelet Transformations", International Journal of Advanced Security in Data Analytics and Networks (Special Issue for Recent Advances in Communications and Networking Technology),2016.

[15] Karthik Sadasivam, Banuprasad Samudrala, T. Andrew Yang. "Design of Network Security Projects using Honeypots", University of Houston.

[16] "Honeypots: Catching the Insider Threat", available at Lance Spitzner Honeypot Technologies Inc. lance@Honeypots.com.

[17] Jyatiti Mokube, Michele Adams, "Honeypots: Concepts, Approaches, and Challenges", Department of Computer Science, Armstrong Atlantic State University.

[18] L. Spitzner, "Honeypots: Tracking Hackers," Boston, USA: Addison Wesley, Parson Education, ISBN 0 321108957, 2003.

[19] Navneet Kambow, Lavleen Kaur Passi, "Honeypots: The Need of Network Security", International Journal of Computer Science and Information Technologies, Vol. 5 (5), 2014.